Incorporating Temporal Constraints in the Analysis Task of a Hybrid Intelligent IDS

Martí Navarro[1], Emilio Corchado [2], Vicente Julián[1] and Álvaro Herrero[3]

Abstract This paper presents an extension of MOVICAB-IDS, a Hybrid Intelligent Intrusion Detection System characterized by incorporating temporal control to enable real-time processing and response. The original formulation of MOVICAB-IDS combines different Computational Intelligence techniques within a multiagent system to perform Intrusion Detection in dynamic computer networks. This work extends the initial proposal by incorporating temporal constraints in the analysis step of the Intrusion Detection process, when a neural projection model is applied.

Keywords Multiagent Systems, Hybrid Artificial Intelligent Systems, Computer Network Security, Intrusion Detection, Temporal Constraints, Time Bounded Deliberative Process.

1 Introduction

Computational Intelligence (CI) has been widely used to build Intrusion Detection Systems (IDSs) [1]. MOVICAB-IDS (MObile VIsualisation Connectionist Agent-Based IDS) has been proposed [2, 3] as a novel IDS employing CI techniques to monitor the network activity. Different CI paradigms are combined to visualise network traffic for Intrusion Detection (ID) at packet level. This intelligent IDS is based on a dynamic Multiagent System (MAS) [4], which integrates an unsupervised neural projection model and the Case-Based Reasoning (CBR) paradigm [5]

[1] Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia, Camino de Vera s/n, 46022, Valencia, Spain. {mnavarro,vinglada}@dsic.upv.es

[2] Departamento de Informática y Automática, Universidad de Salamanca, Plaza de la Merced s/n 37008, Salamanca, Spain. escorchado@usal.es

[3] Civil Engineering Department, University of Burgos. C/ Francisco de Vitoria s/n, 09006 Burgos, Spain. ahcosio@ubu.es

through the use of deliberative agents that are capable of learning and evolving with the environment. A dynamic multiagent architecture is proposed in this study that incorporates both reactive and deliberative (CBR-BDI agents [6]) types of agents. The proposed IDS applies an unsupervised neural projection model [7] to extract interesting traffic dataset projections and to display them through a mobile visualisation interface.

Present approaches involve the application of CI techniques in Real-Time (RT) environments to provide RT systems with 'intelligent' methods to solve complex problems. There are various proposals to adapt CI techniques to RT requirements; the most well-known and promising algorithms within this field being Anytime [8] and approximate processing [9]. One line of research of systems of this kind is related to large applications or hybrid system architectures that embody RT concerns in many components [9], such as Guardian[10], Phoenix [11], or SA-CIRCA [12].

Response time [13] is a critical issue when dealing with security issues. The importance of a smart response on time increases in the case of IDSs. Systems that require a response before a specific deadline, as determined by the system needs, make it essential to monitor execution times. Each task must be performed by the system within a predictable timeframe, within which accurate execution of the given response must be guaranteed. This is the main reason for time-bounding the different tasks of MOVICAB-IDS. The performance of MOVICAB-IDS could be notably improved by integrating RT restrictions. Previous work has incorporated temporal constraints to the planning task, that is, the assignation of each pending analysis to available ‘Analyzer agents’ by the Coordinator agent. This work addresses the incorporation of temporal constraints to the analytical tasks of such IDS. Accordingly, temporal constraints are incorporated in the Analyzer agents maintaining their deliberative capabilities. To do so, the deliberative process relying on a neural projection model is modified to comply with those temporal constraints.

This paper is organized as follows. Section 2 briefly outlines the architecture of MOVICAB-IDS. Section 3 shows how the Analyzer agents in MOVICAB-IDS are upgraded to complete an assigned analysis before a certain deadline. Section 4 presents experimental results to show the benefits that arise from subjecting different phases of CBR to temporal constraints. Finally, the conclusions and future work are discussed in Section 5.

2 MOVICAB-IDS

As proposed for traffic management [14], different tasks perform traffic monitoring and ID. For the data collecting task, a 4-stage framework [15] is adapted to MOVICAB-IDS in the following way: (i) **Data capture:** as network-based ID is pursued, the continual data flow of network traffic must be managed. This data flow contains information on all the packets travelling along the network to be monitored; (ii) **Data selection**: NIDSs have to deal with the practical problem of high volumes of quite diverse data [16]. To manage high diversity of data, MOVICAB-IDS splits the traffic into different groups, taking into account the protocol (UDP, TCP, ICMP, and so on) over IP, as there are differences between the headers of these protocols. Once the captured data is classified by the protocol, it can be processed in different ways; (iii) **Segmentation**: The two first stages do not deal with the problem of continuity in network traffic data. The CMLHL model (as some other neural models) can not process data "on the fly". To overcome this shortcoming, a way of temporarily creating limited datasets from this continuous data flow is proposed by segmentation; (iv) **Data pre-processing**: Finally, the different datasets (simple and accumulated segments) must be preprocessed before presenting them to the neural model. At this stage, categorical features are converted into numerical ones. This happens with the protocol information; each packet is assigned a previously defined value according to the protocol to which it belongs.

Once the data-collecting task is performed and the data is ready, the MOVICAB-IDS process performs two further tasks: (v) Data analysis: CMLHL is applied to analyse the data. Some other unsupervised models have also been applied to perform this task for comparison purposes; (vi) Visualisation: the projections of simple and accumulated segments are presented to the network administrator for scrutiny and monitoring. One interesting feature of the proposed IDS is its mobility; this visualisation task may be performed on a different device other than the one used for the previous tasks. To improve the accessibility of the system, results may be visualised on a mobile device (such as phones or blackberries), enabling informed decisions to be taken anywhere and at any time. In summary, the MOVICAB-IDS task organisation comprises the six tasks described above.

MOVICAB-IDS has been designed, on the basis of Gaia methodology [17], [18], as a MAS that incorporates the following six agents:

- **Sniffer**: this reactive agent is in charge of capturing traffic data. The continuous traffic flow is captured and split into segments in order to send it through the network for further processing. Finally, the readiness of the data is communicated. One agent of this class is located in each of the network segments that the IDS has to cover (from 1 to *n*).
- **Preprocessor**: after splitting traffic data, the generated segments are preprocessed prior to their analysis. Once the data has been preprocessed, an analysis for this new piece of data is requested.
- **Analyzer**: this is a CBR-BDI agent. It has a connectionist model embedded in the adaptation stage of its CBR system that helps to analyze the preprocessed traffic data. The connectionist model is called Cooperative Maximum Likelihood Hebbian Learning (CMLHL) [7]. This agent generates a solution (or achieves its goals) by retrieving a case and analyzing the new one using a CMLHL network.
- **ConfigurationManager**: the configuration information is important as data capture, data splitting, preprocessing and analysis depend on the values of sev-

eral parameters, such as packets to capture, segment length,... This information is managed by the ConfigurationManager reactive agent, which is in charge of providing this information to the Sniffer, Preprocessor, and Analyzer agents.

- **Coordinator**: There can be several Analyzer agents (from 1 to *m*) but only one Coordinator: the latter being in charge of distributing the analyses among the former. In order to improve the efficiency and perform RT processing, the preprocessed data must be dynamically and optimally assigned. This assignment is performed taking into account both the capabilities of the machines where the Analyzer agents are located and the analysis demands (amount and volume of data to be analysed). As is well known, the CBR life cycle consists of four steps: retrieval, reuse, revision and retention [5].
- **Visualizer**: This is an interface agent. At the very end of the process, the analyzed data is presented to the network administrator (or the person in charge of the network) by means of a functional, mobile visualization interface. To improve the accessibility of the system, the administrator may visualize the results on a mobile device, enabling informed decisions to be taken anywhere and at any time.

3 Time-bounding the MOVICAB-IDS Analyzer Agents

The Analyzer agents of MOVICAB-IDS can be classified as hybrid CBR-BDI deliberative agents. CBR-BDI agents [19] integrate the BDI (Belief-Desire-Intention) software model and the Case-Based Reasoning (CBR) paradigm. They use CBR systems [5] as their reasoning mechanism, which enables them to learn from initial knowledge, to interact autonomously with the environment, users and other agents within the system, and which gives them a large capability for adaptation to the needs of its surroundings. MOVICAB-IDS Analyzer agents employ CBR to tune the parameters of the neural model (CMLHL) to analyse preprocessed traffic data. This agent generates a solution (or achieves its goals) by retrieving a previously analysed case and analysing the new one through the CMLHL architecture.

The Analyzer agents incorporate two different modes, namely "learning" and "exploitation". Initially, during the set-up stage, this agent incorporates new knowledge (modelled as sets of problem/solution) into the case base by following the four stages of the CBR cycle. Once the case base is wide enough, the exploitation mode starts. From then on, the revise and retain stages of the CBR cycle are no longer performed. When a new analysis request arrives, the Analyzer agent retrieves the most similar case previously stored in the case base. Then, the weights contained in the solution are reused to project the new data.

The Analyzer is clearly the most resource-consuming class of MOVICAB-IDS agents while they are training the CMLHL neural model (in the learning mode). The amount of computational resources needed to analyze the data coming from different network segments is extremely high. To overcome this demand, Analyzer agents can be located in high-performance computing clusters or in less powerful machines whose computing resources are under-used. In this way, a temporal bounded version of MOVICAB-IDS can be adapted to the available resources for intrusion detection. Additionally, a temporal bounded version of this will cause a reduction in the response time (especially in the worst case) while reducing the amount of considered solutions. As a consequence, less trainings of the neural model will be performed during the learning mode, but on the other hand, it will ensure that these agents are capable of obtaining a result in a fast and predictable way.

To do this, the Analyzer Agent has been upgraded using a Temporal Bounded CBR approach, assuring a temporal bounded behavior in all of its phases. The main improvement is the re-definition of the learning phase as an anytime algorithm, in such a way that the result of the training will be improved when extra time is available to complete this phase.

3.1 Temporal Bounded CBR

In RT environments, the CBR stages must be temporal bounded to ensure that the solutions are produced on time; giving the system a temporal bounded deliberative case-based behaviour. So, the Temporal Bounded CBR (TB-CBR) is a modification of the classic CBR cycle specially adapted to be applied in domains with temporal constraints.

The different phases of the TB-CBR cycle are grouped in two stages according to their function within the reasoning process of an agent with RT constraints. The fist one, called learning stage, consists of the revise and retain phases; and the second one, named the deliberative stage, includes the retrieve and reuse phases. Each phase will schedule its own execution time to support the designer in the time distribution among the TB-CBR phases. These stages are modelled as anytime algorithms [20], where the process is iterative and each time-bounded iteration may improve the final response.

To ensure up-to-date cases in the case base, the TB-CBR cycle starts at the learning stage, which entails checking whether previous cases are awaiting revision and could be stored in the case base. The solutions provided by the TB-CBR are stored in a solution list at the end of the deliberative stage. This list is accessed when each new TB-CBR cycle begins. If there is enough time, the learning stage is implemented for cases where solution feedback has recently been received. If the list is empty, this process is omitted.

Once the learning stage finishes, the deliberative stage starts. The retrieval algorithm is used to search the case base and chose a case that is similar to the current case (i.e. the one that characterizes the problem to be solved). Each time a similar case is found, it is sent to the reuse phase where it is transformed into a suitable plan for the current problem by using a reuse algorithm. Therefore, at the end of each iteration in the deliberative stage, the TB-CBR method is able to provide a solution to the problem at hand, which may be improved in subsequent iterations if there is any time remaining at the deliberative stage. A more detailed explanation of the TB-CBR algorithm can be seen in [21].

3.2 Integrating TB-CBR into the MOVICAB-IDS Analyzer Agent

As discussed in previous sections, the Analyzer agent has two modes of behavior as can be seen in Fig. 1. The first one (learning mode) is executed by the agent at the beginning of its life cycle in order to incorporate new knowledge while analyzing the packets of new analysis requests. Due to the requirements to this mode, it is necessary to run the whole TB-CBR cycle (deliberative + learning stages). Nevertheless, when the agent has enough knowledge (cases in the case-base), it switches to the exploitation mode. In this mode, the agent only has to run the deliberative stage of the TB-CBR algorithm. The TB-CBR algorithm allows the system to dynamically change the time assigned for each one of the two stages. So, starting with the exploitation mode is simply to assign all the available time to the deliberative stage of the TB-CBR algorithm.

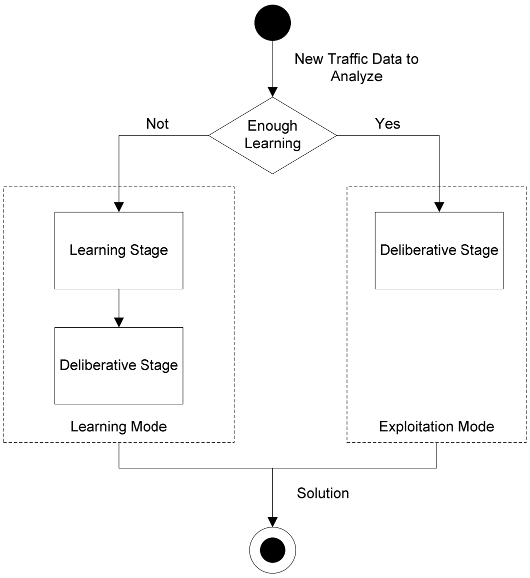

Fig. 1. Analyzer agent modes.

The techniques and tools applied in the CBR cycle are redefined through the TB-CBR method as follows:

- **Retrieve and Reuse phases (Deliberative stage)**: these phases of the TB-CBR are always run, whatever the mode of Analyzer agent. In this way, when a new analysis is requested, the Analyzer agent tries to find the most similar case to the new one in the case base and it is reused to obtain a solution (the values of the parameters used to train the CMLHL model). Theses phases are implemented by means of the *anytime* algorithm. This algorithm extracts a solution in a known amount of time, smaller than the one available to complete these phases. In the reuse phase, a set of trainings for the CMLHL neural model are defined by combining the different parameter values recovered from the cases in the case base. As the number of iterations of each one of them is known, the training time is also known. As a result, the Analyzer agent can predict how many neural network models could be built in the available time. The available time of the Analyzer agent to complete the required analysis will be greater when working in the exploitation mode. This is because the anytime behavior of the algorithm will perform better the longer the available time will be.
- **Revise and Retain phases (Learning stage)**: As the revise and retain phases depend of the human experience, these phases are completed offline. Once the human expert performs a visual analysis of the segment, one of the projections is selected and the associated parameters are stored in the case-base to be considered in future executions. The time required by a human expert to perform this action is variable and indeterminate. As a consequence, these phases are outside the real-time decision algorithm used by the Analyzer agents.

4 Experimental Results

The effectiveness of MOVICAB-IDS in facing some anomalous situations has been widely demonstrated in previous works [2], [3], [22]. It identifies anomalous situations due to the fact that these situations do not tend to resemble parallel and smooth directions (normal situations) or because their high temporal concentration of packets.

As this work addresses the real-time issue, some experiments dealing with such feature of the Analyzer agents have been carried out. The main idea in this experimental study is to check how the application of the RT-CBR cycle modifies the performance of those agents in terms of the output projection for the learning mode. To do so, for each dataset, the projection selected by the network administration (as the most informative one) has been tracked. The following data are provided in Table 1:

- **Deadline**: amount of time allocated for the segment analysis (deliberative stage).
- **Success rate**: percentage of segment analysis including the most informative projection.

- **Average number of parameter combinations**: amount of combinations generated after the case reuse.
- **Average number of performed ANN trainings**: amount of ANN trainings that have been performed by selecting some of the parameter combinations due to the temporal constraints.

Table 1. Performance of the Analyzer agents incorporating TB-CBR.

Deadline (s)	Success rate	Average number of parameter combinations	Average number of performed ANN trainings
10	73%	361	9
60	85%	361	48
120	99%	361	103

5 Conclusions

An upgraded version of MOVICAB-IDS, incorporating temporal constraints in the Analyzer agents, is presented in this paper. Temporal constraints are imposed on the deliberative agents within a CBR architecture, which enables those agents to respond to requested analysis in real (both hard or soft) time. To do so the four phases of the CBR cycle are redefined. The consequences of temporal bounding these phases are described in this paper. As a result, the Analyzer agents will always generate a projection within the available time. As empirically checked, this time limitation does not imply lower quality of the provided solution.

Acknowledgments: This research is funded through the Junta of Castilla and León project BU006A08; projects of the Spanish Ministry of Science and Innovation CIT-020000-2009-12 and TIN2010-21272-C02-01; the Spanish government (TIN2009-13839-C03-01); the Generalitat Valenciana (ROMETEO 2008/051), FEDER (European Regional Development Fund) and CONSOLIDER-INGENIO (2010 CSD2007-00022). The authors would also like to thank the vehicle interior manufacturer, Grupo Antolin Ingenieria S.A. for supporting the project through the MAGNO2008 - 1028.- CENIT project funded by the Spanish Ministry of Science and Innovation**.**

References

1. Abraham, A., Jain, R., Thomas, J., Han, S.Y. (2007) D-SCIDS: Distributed Soft Computing Intrusion Detection System. Journal of Network and Computer Applications 30(1): 81-98
2. Herrero, Á., Corchado, E. (2009) Mining Network Traffic Data for Attacks through MOVICAB-IDS. (ed) Foundations of Computational Intelligence 4. Springer. Studies in Computational Intelligence 377-394

3. Corchado, E., Herrero, Á. (2010) Neural Visualization of Network Traffic Data for Intrusion Detection. Applied Soft Computing ("Accepted - In press")
4. Wooldridge, M., Jennings, N., R. (1995) Agent theories, architectures, and languages: A survey. Intelligent Agents
5. Aamodt, A., Plaza, E. (1994) Case-Based Reasoning - Foundational Issues, Methodological Variations, and System Approaches. AI Communications 7(1): 39-59
6. Carrascosa, C., Bajo, J., Julián, V., Corchado, J.M., Botti, V. (2008) Hybrid Multi-agent Architecture as a Real-Time Problem-Solving Model. Expert Systems with Applications: An International Journal 34(1): 2-17
7. Corchado, E., Fyfe, C. (2003) Connectionist Techniques for the Identification and Suppression of Interfering Underlying Factors. International Journal of Pattern Recognition and Artificial Intelligence 17(8): 1447-1466
8. Dean, T., Boddy, M. (1988) An Analysis of Time-dependent Planning. (ed) 7th National Conference on Artificial Intelligence.
9. Garvey, A., Lesser, V. (1994) A Survey of Research in Deliberative Real-time Artificial Intelligence. Real-Time Systems 6(3): 317-347
10. Hayes-Roth, B., Washington, R., Ash, D., Collinot, A., Vina, A., Seiver, A. (1992) Guardian: A Prototype Intensive-care Monitoring Agent. Artificial Intelligence in Medicine 4: 165-185
11. Howe, A.E., Hart, D.M., Cohen, P.R. (1990) Addressing Real-time Constraints in the Design of Autonomous Agents. Real-Time Systems 2(1): 81-97
12. Musliner, D.J., Durfee, E.H., Shin, K.G. (1993) CIRCA: A Cooperative Intelligent Real-time Control Architecture. IEEE Transactions on Systems, Man, and Cybernetics 23(6): 1561 - 1574
13. Kopetz, H. (1997) Real-time Systems: Design Principles for Distributed Embedded Applications. Kluwer Academic Publishers
14. Babu, S., Subramanian, L., Widom, J. (2001) A Data Stream Management System for Network Traffic Management. (ed) Workshop on Network-Related Data Management (NRDM 2001).
15. Herrero, Á., Corchado, E. (2008) Traffic Data Preparation for a Hybrid Network IDS. (ed) Third International Workshop on Hybrid Artificial Intelligence Systems (HAIS 2008) 5271. Springer, Heidelberg. LNAI
16. Dreger, H., Feldmann, A., Paxson, V., Sommer, R. (2004) Operational Experiences with High-Volume Network Intrusion Detection. (ed) 11th ACM Conference on Computer and Communications Security. ACM Press New York.
17. Zambonelli, F., Jennings, N.R., Wooldridge, M. (2003) Developing Multiagent Systems: the Gaia Methodology. ACM Transactions on Software Engineering and Methodology 12(3): 317-370
18. Wooldridge, M., Jennings, N.R., Kinny, D. (2000) The Gaia Methodology for Agent-Oriented Analysis and Design. Autonomous Agents and Multi-Agent Systems 3(3): 285-312
19. Pellicer, M.A., Corchado, J.M. (2005) Development of CBR-BDI Agents. International Journal of Computer Science and Applications 2(1): 25 - 32
20. Dean, T., Boddy, M.S. (1988) An Analysis of Time-Dependent Planning. (ed) 7th National Conference on Artificial Intelligence.
21. Navarro, M., Heras, S., Julián, V. (2009) Guidelines to Apply CBR in Real-Time Multi-Agent Systems. Journal of Physical Agents 3(3): 39-43
22. Corchado, E., Herrero, Á., Sáiz, J.M. (2005) Detecting Compounded Anomalous SNMP Situations Using Cooperative Unsupervised Pattern Recognition. In: Duch, W., Kacprzyk, J., Oja, E., Zadrozny, S. (ed) 15th International Conference on Artificial Neural Networks (ICANN 2005) 3697. Springer, Heidelberg. LNCS